希腊值的推导与实现

杨宸宇

2016301550186

我们已知看涨期权定价公式为:

$$C = SN(d_1) - Ke^{-r(T-t)}N(d_2) \tag{1}$$

其中:

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}} \tag{2}$$

$$d_2 = \frac{\ln(S/K) + (r - \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{(T-t)} \tag{4}$$

$$N(\alpha) = \int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \tag{5}$$

然后以此为基础对 5 个希腊值进行推导

1 Delta

Delta 是 C 对 S 的一阶偏导数，于是我们有如下证明：

$$\frac{\partial C}{\partial S} = N(d_1) + S * \frac{\partial N(d_1)}{\partial S} - Ke^{r(T-t)} * \frac{\partial N(d_2)}{S} \tag{6}$$

$$= N(d_1) + S * \frac{\partial N(d_1)}{\partial S} - Ke^{r(T-t)} * \frac{\partial N(d_2)}{S} \tag{7}$$

我们可以得到:

$$\begin{aligned}
\frac{\partial N(d_1)}{\partial S} &= \frac{1}{\sqrt{2\pi}} e^{-\frac{d_1^2}{2}} * d_1' && (8)\\
\frac{\partial N(d_2)}{\partial S} &= \frac{1}{\sqrt{2\pi}} e^{-\frac{d_2^2}{2}} * d_2' && (9)\\
\frac{\partial d_1}{\partial S} &= \frac{1}{\sigma\sqrt{(T-t)}} * \frac{K}{S} * \frac{1}{K}\\
&= \frac{1}{\sigma S\sqrt{(T-t)}} && (10)\\
\frac{\partial d_2}{\partial S} &= \frac{1}{\sigma S\sqrt{(T-t)}} && (11)
\end{aligned}$$

因此，原始式子转换成:

$$\frac{\partial C}{\partial S} = N(d_1) + \frac{1}{\sigma\sqrt{2\pi(T-t)}} e^{-\frac{d_1^2}{2}} - \frac{Ke^{-r(T-t)}}{\sigma S\sqrt{2\pi(T-t)}} e^{-\frac{d_2^2}{2}} \tag{12}$$

计算 d_1^2 和 d_2^2，我们得到:

$$d_1^2 = \frac{\ln^2(S/K) + (r+\sigma^2/2)^2(T-t)^2 + 2\ln(S/K)(r+\sigma^2/2)(T-t)}{\sigma^2(T-t)} \tag{13}$$

$$d_2^2 = \frac{\ln^2(S/K) + (r-\sigma^2/2)^2(T-t)^2 + 2\ln(S/K)(r-\sigma^2/2)(T-t)}{\sigma^2(T-t)} \tag{14}$$

发现其中有公共因子，提取出来，令:

$$A = \frac{1}{\sigma\sqrt{2\pi(T-t)}} e^{-\frac{\ln^2(S/K)+(r^2+\sigma^4/4)(T-t)^2}{2\sigma^2(T-t)}} \tag{15}$$

于是，我们可以见 (12)中的后两项进行化简:

$$B = \frac{1}{\sigma\sqrt{2\pi(T-t)}} e^{-\frac{d_1^2}{2}} - \frac{Ke^{-r(T-t)}}{\sigma S\sqrt{2\pi(T-t)}} e^{-\frac{d_2^2}{2}} \tag{16}$$

$$\begin{aligned}
B &= A * (e^{(\frac{-r\sigma^2(T-t)+2\ln(S/K)+(r+\sigma^2/2)}{\sigma^2})} - \frac{Ke^r(T-t)}{S} e^{(\frac{r\sigma^2(T-t)+2\ln(S/K)-(r+\sigma^2/2)}{\sigma^2})})\\
&= A * (e^{-\frac{r(T-t)}{2}} * e^{-\frac{\ln(S/K)}{2}} - \frac{Ke^{-r(T-t)}}{S} e^{\frac{r(T-t)}{2}} * e^{\frac{\ln(S/K)}{2}})\\
&= A * (e^{-\frac{r(T-t)}{2}} * e^{-\frac{\ln(S/K)}{2}} - (S/K)^{-1} * e^{\frac{-r(T-t)}{2}} * e^{\frac{\ln(S/K)}{2}})\\
&= A * (e^{-\frac{r(T-t)}{2}} * e^{-\frac{\ln(S/K)}{2}} - e^{-\frac{r(T-t)}{2}} * e^{-\frac{\ln(S/K)}{2}})\\
&= 0 && (17)
\end{aligned}$$

因此我们最终得到：

$$\boldsymbol{\Delta} = N(d_1) \tag{18}$$

同时(12)后两项相等，我们得到一个结论，之后会用到：

$$SN'(d_1) = Ke^{-r(T-t)}N'(d_2) \tag{19}$$

其中：

$$N'(d_1) = \frac{1}{\sqrt{2\pi}}e^{-\frac{d_1^2}{2}} \tag{20}$$

$$N'(d_2) = \frac{1}{\sqrt{2\pi}}e^{-\frac{d_2^2}{2}} \tag{21}$$

2 Gamma

Gamma 是 C 对 S 的二阶导数，即 Delta 对 S 的一阶导数，我们有如下证明：

$$\begin{aligned}
\frac{\partial\delta}{\partial S} &= \frac{\partial N(d_1)}{\partial S} \\
&= \frac{1}{\sqrt{2\pi}}e^{-\frac{d_1^2}{2}} * \frac{\partial d_1}{\partial S} \\
&= \frac{1}{\sqrt{2\pi}}e^{-\frac{d_1^2}{2}} * \frac{1}{\sigma S\sqrt{T-t}} \\
&= \frac{1}{\sqrt{2\pi(T-t)}\sigma S}e^{-\frac{d_1^2}{2}}
\end{aligned} \tag{22}$$

所以我们得到：

$$\boldsymbol{\Gamma} = \frac{1}{\sigma S\sqrt{2\pi(T-t)}}e^{-\frac{d_1^2}{2}} \tag{23}$$

也可以表示为:

$$\boldsymbol{\Gamma} = \frac{N'(d_1)}{S\sigma\sqrt{T-t}} \tag{24}$$

3 Theta

Theta 是 C 对 t 的一阶导数，我们有如下证明：

$$\frac{\partial C}{\partial t} = N'(d_1) - Ke^{-r(T-t)}rN(d_2) - Ke^{-r(T-t)}N'(d_2)\frac{\partial d_2}{\partial t} \tag{25}$$

我们根据 (19)，可以对(25)进行化简：

$$\frac{\partial C}{\partial t} = -rKe^{-r(T-t)}N(d_2) + SN'(d_1)\left(\frac{\partial d_1}{\partial t} - \frac{\partial d_2}{\partial t}\right) \tag{26}$$

根据 d_1 和 d_2 的等价关系(4)可以知道：

$$d_1 - d_2 = \sigma\sqrt{T-t} \tag{27}$$

$$\begin{aligned}\frac{\partial d_1}{\partial t} - \frac{\partial d_2}{\partial t} &= \frac{\partial(\sigma\sqrt{T-t})}{\partial t} \\ &= -\frac{\sigma}{2\sqrt{T-t}}\end{aligned} \tag{28}$$

所以我们最终得到：

$$\frac{\partial C}{\partial t} = -rKe^{-r(T-t)}N(d_2) - SN'(d_1)\frac{\sigma}{2\sqrt{T-t}} \tag{29}$$

$$\boldsymbol{\Theta} = -rKe^{-r(T-t)}N(d_2) - SN'(d_1)\frac{\sigma}{2\sqrt{T-t}} \tag{30}$$

4 Vega

Vega 是 C 对 σ 的一阶导数，我们有如下证明：

$$\frac{\partial C}{\partial \sigma} = SD'(d_1)\frac{\partial d_1}{\partial \sigma} - Ke^{-r(T-t)}N(d_2)\frac{\partial d_2}{\partial \sigma} \tag{31}$$

同样根据(4)和(19)：

$$\frac{\partial C}{\partial \sigma} = SN'(d_1)(\frac{\partial d_1}{\partial t} - \frac{\partial d_2}{\partial t}) \tag{32}$$

$$\frac{\partial d_1}{\partial t} - \frac{\partial d_2}{\partial t} = \sqrt{T-t} \tag{33}$$

所以我们最终计算出：

$$\frac{\partial C}{\partial \sigma} = SN'(d_1)\sqrt{T-t} \tag{34}$$

$$\boldsymbol{V} = SN'(d_1)\sqrt{T-t} \tag{35}$$

5 Rho

Rho 是 C 对 r 的一阶导数，我们有如下证明：

$$\frac{\partial C}{\partial r} = SN'(d_1)\frac{\partial d_1}{\partial r} + Ke^{-r(T-t)}(T-t)N(d_2) - Ke^{-r(T-t)}N'(d_2)\frac{\partial d_2}{\partial r} \tag{36}$$

根据等价关系(4)和(19):

$$\begin{aligned} SN'(d_1)\frac{\partial d_1}{\partial r} - Ke^{-r(T-t)}N'(d_2)\frac{\partial d_2}{\partial r} &= SN'(d_1)(\frac{\partial d_1}{\partial r} - \frac{\partial d_2}{\partial r}) \\ &= SN'(d_1) * 0 \\ &= 0 \end{aligned} \tag{37}$$

所以，我们最终得到：

$$\frac{\partial C}{\partial r} = (T-t)Ke^{-r(T-t)}N(d_2) \tag{38}$$

$$\boldsymbol{\rho} = (T-t)Ke^{-r(T-t)}N(d_2) \tag{39}$$

6 Python 代码实现

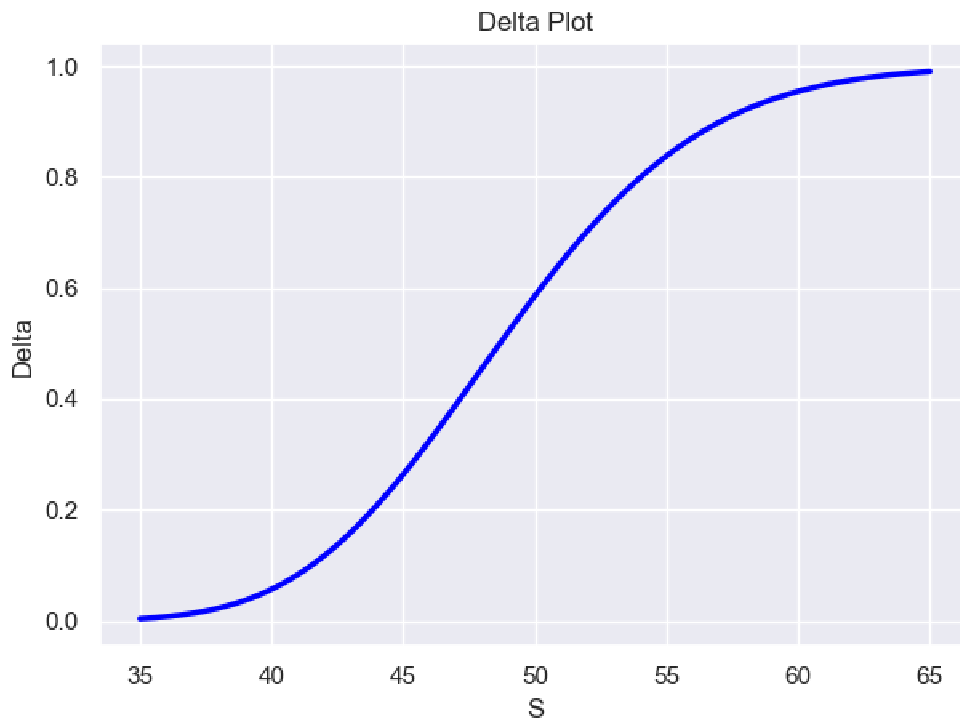
```python
# -*-coding:utf-8-*-

# 主题: 第二次金融工程学实验作业
# 作者: 杨宸宇
# 学号: 2016301550186
# 日期: 11\28\2018

# 调用所需要的库
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from math import log, sqrt, exp, pi
from scipy import stats


class Greek(object):

    def __init__(self, S0, K, T, r, sigma):
        '''

        :param S0: 标的价格
        :param K: 期权的执行价格
        :param T: 时间
        :param r: 无风险利率
        :param sigma: 波动率

        '''
        self.S0 = S0
        self.K = K
        self.T = T
        self.r = r
        self.sigma = sigma
        self.d1 = (log(self.S0 / self.K) + (self.r + 0.5 *
            self.sigma ** 2) * self.T) / (self.sigma *
            sqrt(self.T))
```

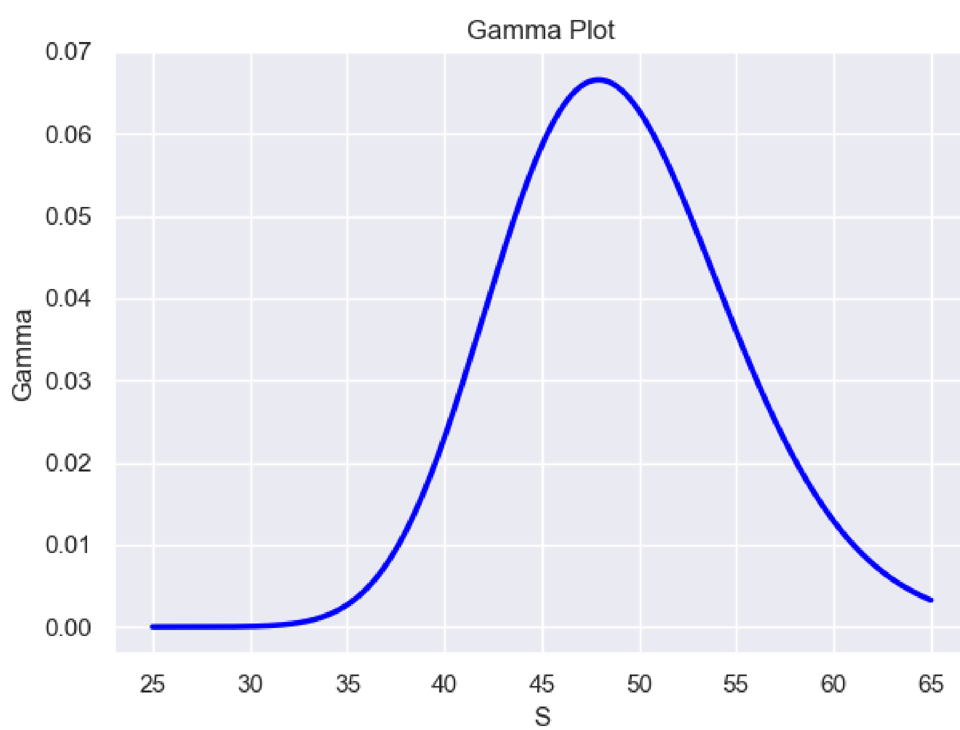
```python
        self.d2 = (log(self.S0 / self.K) + (self.r - 0.5 *
            self.sigma ** 2) * self.T) / (self.sigma *
            sqrt(self.T))

    def bs_call_value(self):
        value = (self.S0 * stats.norm.cdf(self.d1, 0., 1.)) \
              - self.K * exp(-self.r * self.T) *
                  stats.norm.cdf(self.d2, 0., 1.)
        return value

    def delta(self):
        value = stats.norm.cdf(self.d1, 0., 1.)
        return value

    def gamma(self):
        value = 1 / (sqrt(2 * pi * self.T) * self.sigma * self.S0)
            * exp(-self.d1 ** 2 / 2)
        return value

    def theta(self):
        value = -self.r * self.K * exp(-self.r * self.T) *
            stats.norm.cdf(self.d2, 0., 1.) \
              - (self.S0 * 1 / sqrt(2 * pi) * exp(-self.d1 ** 2 /
                  2) * self.sigma) / (2 * sqrt(self.T))
        return value

    def vega(self):
        value = self.S0 * sqrt(self.T) * 1 / sqrt(2 * pi) *
            exp(-self.d1 ** 2 / 2)
        return value

    def rho(self):
        value = self.T * self.K * exp(-self.r * self.T) *
            stats.norm.cdf(self.d2, 0., 1.)
        return value


# 实验函数结果------------------------------------------
greek = Greek(49, 50, 0.3846, 0.05, 0.2)
```

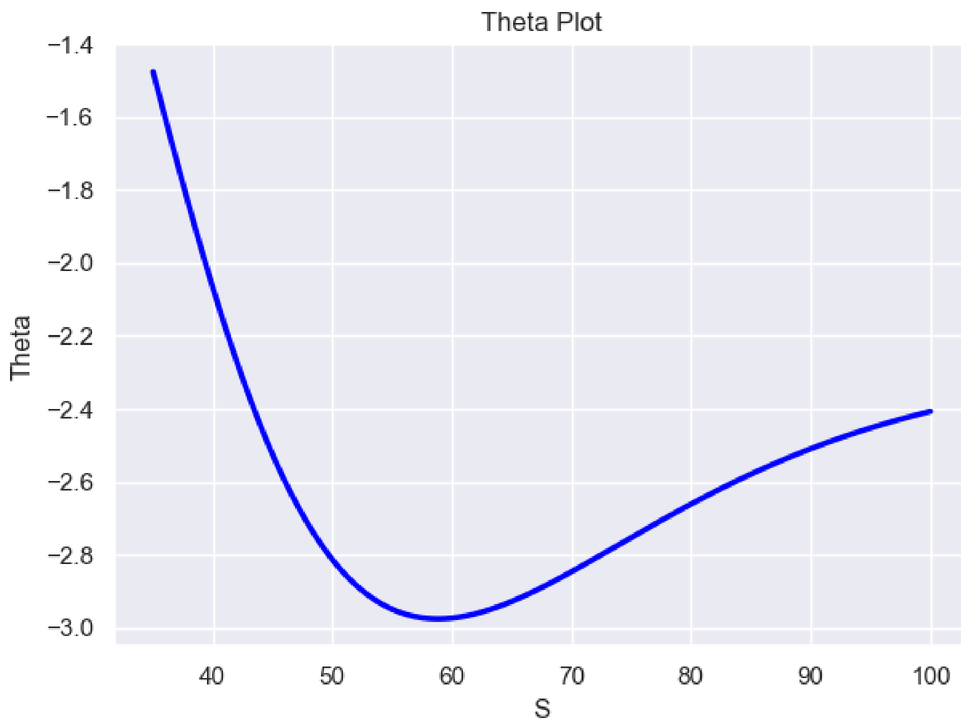
```
print("The Delta is %f\nThe Gamma is %f\nThe Theta is %f\nThe
    Vega is %f\nThe Rho is %f" % (
greek.delta(), greek.gamma(), greek.theta(), greek.vega(),
    greek.rho()))

# 生成各个希腊值的曲线
# Delta
data = pd.DataFrame()
data['S'] = np.linspace(35, 65, 1000)
data['Delta'] = np.nan
for i in range(len(data)):
    data.iloc[i, 1] = Greek(data.iloc[i, 0], 50, 0.3846, 0.05,
        0.2).delta()
sns.set()
p = sns.lineplot(x='S', y='Delta', data=data, color='blue',
    linewidth=2.5)
p.set_title('Delta Plot')
plt.show()

# Gamma
data = pd.DataFrame()
data['S'] = np.linspace(25, 65, 1000)
data['Gamma'] = np.nan
for i in range(len(data)):
    data.iloc[i, 1] = Greek(data.iloc[i, 0], 50, 0.3846, 0.05,
        0.2).gamma()
sns.set()
p = sns.lineplot(x='S', y='Gamma', data=data, color='blue',
    linewidth=2.5)
p.set_title('Gamma Plot')
plt.show()

# Theta
data = pd.DataFrame()
data['S'] = np.linspace(35, 100, 1000)
data['Theta'] = np.nan
for i in range(len(data)):
    data.iloc[i, 1] = Greek(data.iloc[i, 0], 50, 2, 0.05,
        0.25).theta()
```

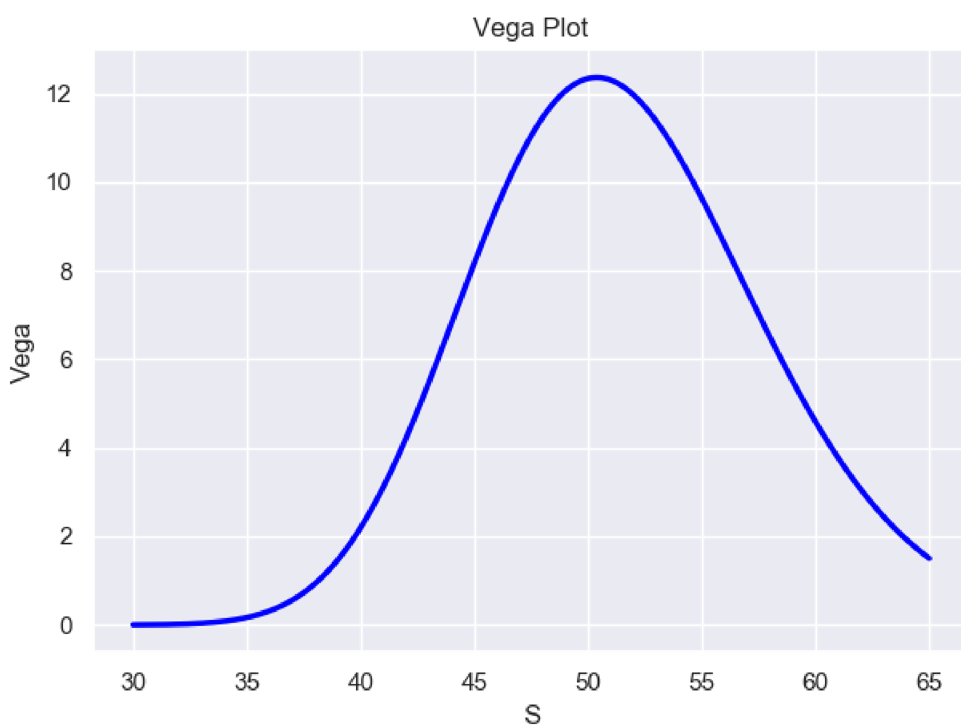
```python
sns.set()
p = sns.lineplot(x='S', y='Theta', data=data, color='blue',
    linewidth=2.5)
p.set_title('Theta Plot')
plt.show()

# Vega
data = pd.DataFrame()
data['S'] = np.linspace(30, 65, 1000)
data['Vega'] = np.nan
for i in range(len(data)):
    data.iloc[i, 1] = Greek(data.iloc[i, 0], 50, 0.3846, 0,
        0.2).vega()
sns.set()
    p = sns.lineplot(x='S', y='Vega', data=data, color='blue',
        linewidth=2.5)
p.set_title('Vega Plot')
plt.show()

# Rho
data = pd.DataFrame()
data['S'] = np.linspace(35, 65, 1000)
data['Rho'] = np.nan
for i in range(len(data)):
    data.iloc[i, 1] = Greek(data.iloc[i, 0], 50, 0.3846, 0.05,
        0.2).rho()
sns.set()
p = sns.lineplot(x='S', y='Rho', data=data, color='blue',
    linewidth=2.5)
p.set_title('Rho Plot')
plt.show()
```

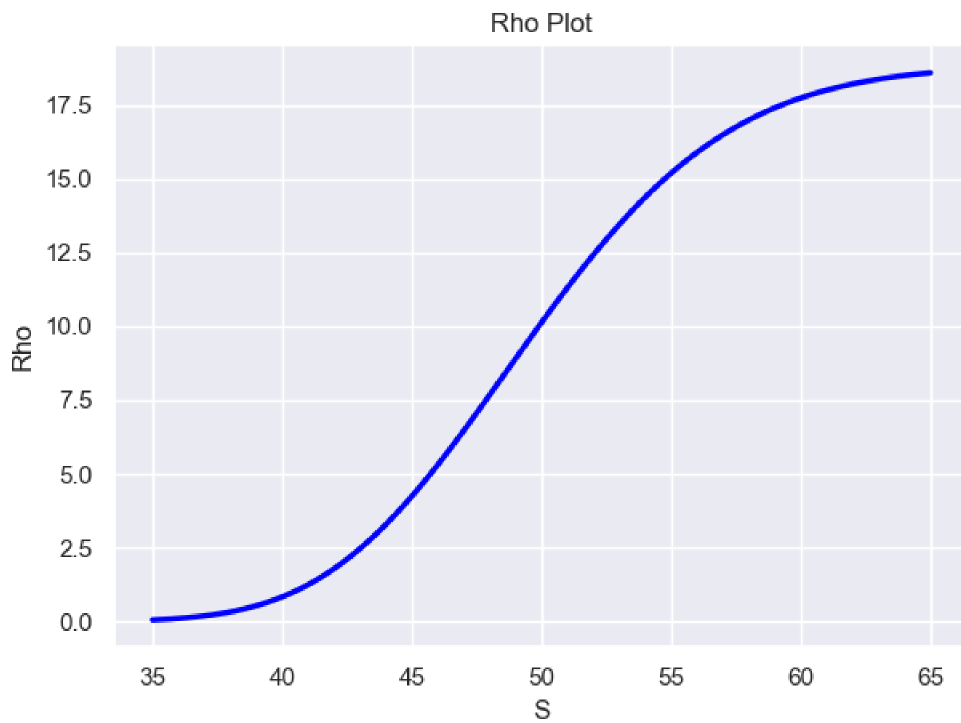
各希腊值和标的资产价格变动关系见后页:


Delta Plot
1.0
0.8
0.6
0.4
0.2
0.0
Delta
35
40
45
50
55
60
65
S


Gamma Plot
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Gamma
25
30
35
40
45
50
55
60
65
S


Theta Plot
Theta
S
−1.4
−1.6
−1.8
−2.0
−2.2
−2.4
−2.6
−2.8
−3.0
40
50
60
70
80
90
100


Vega Plot
Vega
S
0
2
4
6
8
10
12
30
35
40
45
50
55
60
65


Rho Plot
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Rho
35
40
45
50
55
60
65
S